

NFR & Architectuur

Bekeken vanuit de requirements engineer

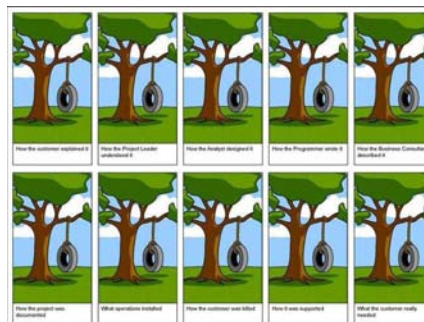
SPider sessie
23 september 2010

Sven van der Zee
sven.van.der.zee@dnv.com

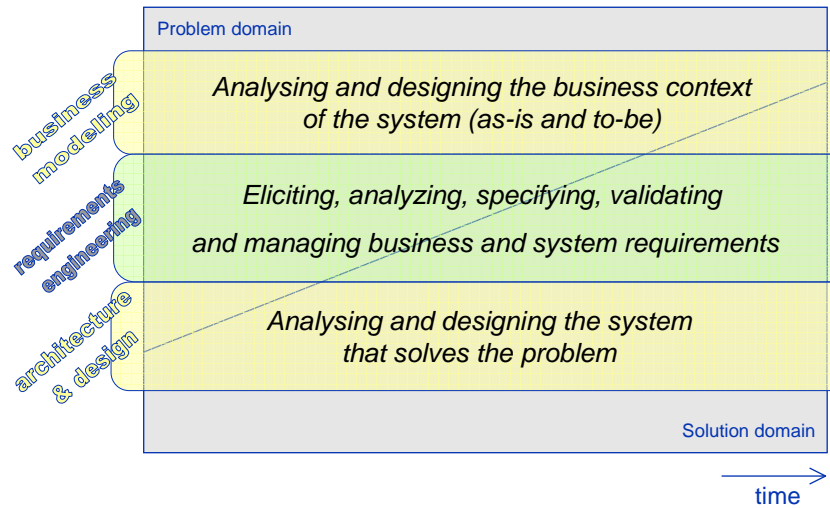


Requirements Engineering

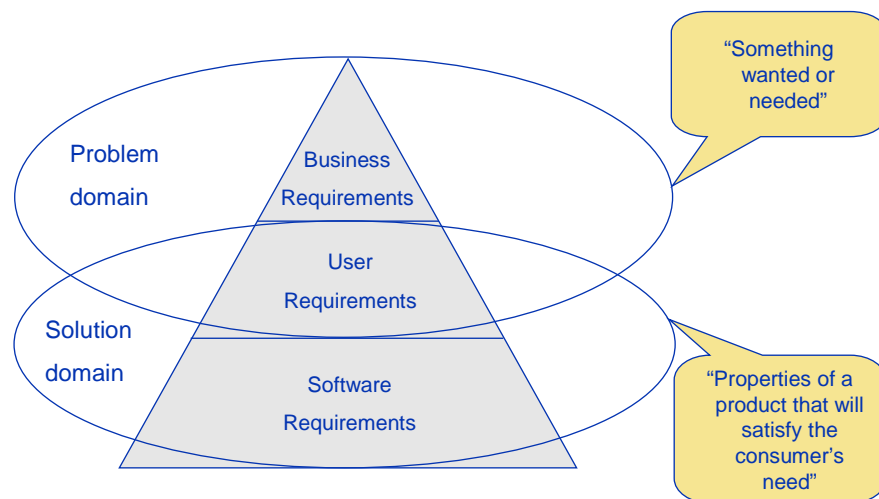
- Requirements engineering is a software engineering discipline with the responsibility to **identify, analyze, document, validate and manage** requirements for the system to be developed.
- The goal of requirements engineering is to build a concrete, **shared understanding** of the problem and its solution among all the stakeholders: sponsor, users, project manager and developers.



Requirements Engineering

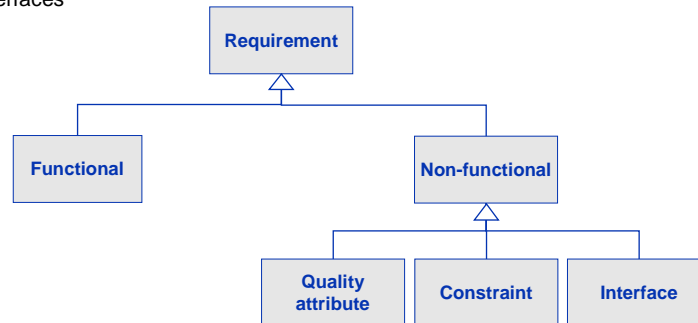


Requirements Levels



Functional and Non-Functional

- Functional requirements define the behavior of a product
 - Things it must do (for its users) or allow the users to do with it
- Non-functional requirements are the other properties:
 - Quality attributes
 - Design and implementation constraints
 - External interfaces

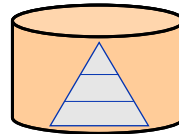


Example

- The system shall ensure that data is protected from unauthorised access.
 - Conventionally, this would be considered as a **non-functional** requirement because it does not specify specific system functionality which must be provided. However, it could have been specified in slightly more detail as follows
- The system shall include a user authorisation procedure where users must identify themselves using a login name and password in order to get access to the system data.
 - In this form, the requirement looks rather more like a **functional** requirement as it specifies a function (user login) which must be incorporated in the system

Requirements Management

- Store all the requirements (with metadata/attributes) in a repository
- Inform stakeholders with information from the repository
- Control changes to requirements
- Control versions of requirements
- Establish baselines

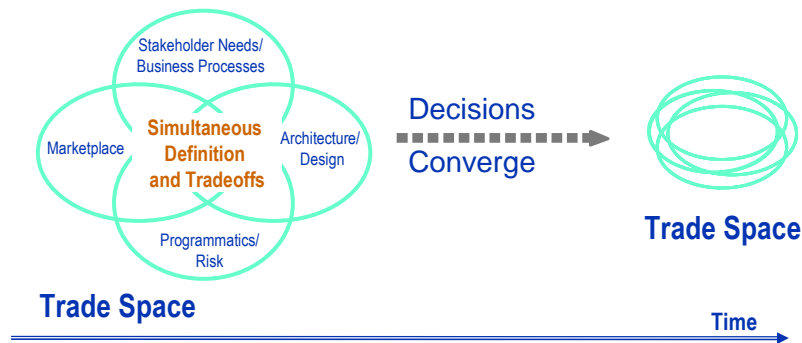


Requirements Analysis

- Requirements analysis checks (all) requirements for:
 - Necessity
 - Consistency
 - Completeness
 - Feasibility
- Conflicts and issues in (all) requirements are resolved through prioritization and negotiation with stakeholders

Spheres of Influence

- Business driven systems demand processes that simultaneously define, balance, negotiate, and converge four spheres of influence over time to systematically evolve the definition of the solution to be implemented.

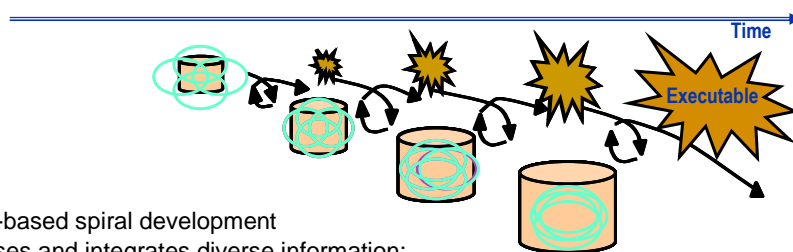


NFR & Architectuur
23 september 2010
© Det Norske Veritas AS. All rights reserved.

8

MANAGING RISK
DNV

Knowledge Grows Incrementally



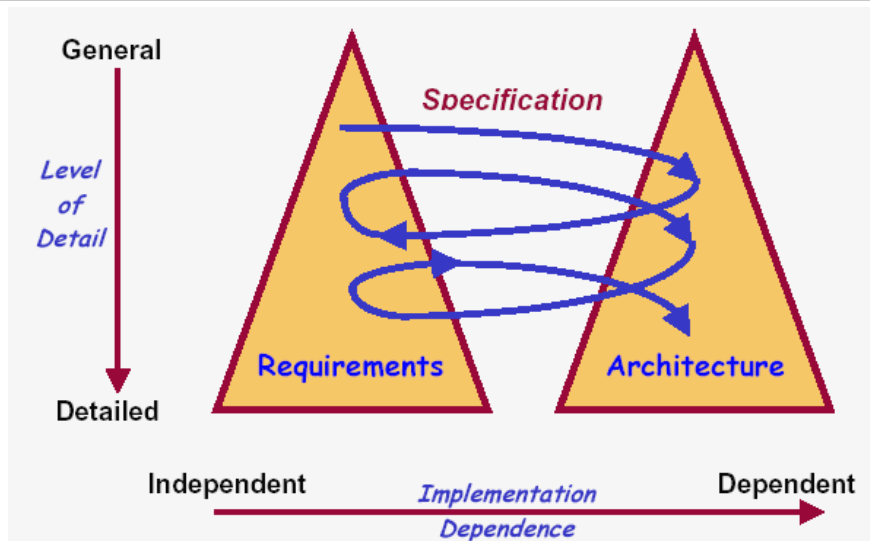
- Risk-based spiral development focuses and integrates diverse information:
 - Prioritized stakeholder needs, end-user processes
 - Organization and system architecture, design constraints
 - Identified risks, project constraints
 - Marketplace offerings, product characteristics, other buyer usage
- Frequent, evolving executable representations demonstrate current understanding

NFR & Architectuur
23 september 2010
© Det Norske Veritas AS. All rights reserved.

9

MANAGING RISK
DNV

Twin Peaks Model



NFR & Architectuur
23 september 2010
© Det Norske Veritas AS. All rights reserved.

10

MANAGING RISK
DNV

Responsibilities

Requirements Engineer must:

- **Identify** all of the requirements
- **Analyze** requirements to ensure that they are good requirements (necessary, complete, consistent, feasible)
- Properly **specify** all of the requirements
- **Validate** requirements with all stakeholders
- **Store** all of the requirements, provide appropriate metadata, **maintain** them under configuration control, and **freeze** them at the appropriate time so that architecture, design, and implementation can be completed and so that the current version of the system can be tested and accepted

Architect must:

- Provide **feedback** about technical feasibility and consequences
- Understand the impact on the architecture and make engineering **trade-offs** between the conflicting requirements in order to ensure an optimized architecture
- Properly **allocate** the architecture-relevant requirements to the associated architectural components

NFR & Architectuur
23 september 2010
© Det Norske Veritas AS. All rights reserved.

11

MANAGING RISK
DNV

Summary

- The requirements engineer is responsible for the development and management of all sorts (functional and non-functional) of requirements on all levels.
- The requirements engineer is positioned as an independent facilitator to bring and keep all stakeholders together in a shared understanding of problem and solution.
- The requirements engineer works closely together with the architect to get and keep the requirements and the architecture aligned.

Safeguarding life, property and the environment

www.dnv.nl

